DOCUMENT RESUME

ED 266 459 CS 209 522

AUTHOR Hellwig, Harold H.
TITLE Developing Expert Systems for the Analysis of Syntactic and Semantic Patterns.
PUB DATE May 85
NOTE 20p.; Paper presented at the Meeting of the UCLA Conference on "Computers and Writing: New Directions in Teaching and Research" (Los Angeles, CA, May 4-5, 1985).
PUB TYPE Viewpoints (120) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Case (Grammar); Computer Assisted Instruction; *Computer Oriented Programs; *Computer Software; *Discourse Analysis; *Programing; Semantics; *Sentence Structure; Syntax; *Writing Instruction
IDENTIFIERS *Expert Systems

ABSTRACT

Noting that expert computer systems respond to various contexts in terms of knowledge representation, this paper explains that heuristic rules of production, procedural representation, and frame representation have been adapted to such areas as medical diagnosis, signal interpretation, design and planning of electrical circuits and computer system configurations, data analysis in protein crystallography and geology, and expert system construction itself. The paper goes on to examine production rules--procedural and frame representations in the development of an expert system that can analyze syntactic and semantic patterns in written discourse. It is observed that case grammar analysis and semantic differential scale techniques, encoded into a computer program called BOSS, generate textual analysis. Finally, ways an expert system can be used in a writing classroom are discussed. (Author/HOD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Harold H. Hellwig

"Developing Expert Systems for the Analysis of Syntactic and Semantic Patterns"

ED266459

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Harold H. Hellwig

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Abstract

The following paper first discusses how expert systems respond to various contexts in terms of knowledge representation. Heuristic rules of production, of procedural representation, and of frame representation have been adapted to such areas as medical diagnosis, signal interpretation, design and planning of electrical circuits and computer system configurations, data analysis in protein crystallography and geology, and expert system construction itself. Second, production rules, procedural and frame representations in the development of an expert system that can analyze syntactic and semantic patterns in written discourse are examined. Case grammar analysis and semantic differential scale techniques, encoded into a computer program called BOSS, generates textual analysis. Finally, ways an expert system can operate in the context of the classroom, acting as a surrogate teacher, are discussed.

Introduction

Expert systems have been around for some time under a variety of disguises--in fact, we claim to be expert writers showing our students how to write in some kind of real world out there. Perhaps a few of us have taught a class without extensive experience in some field--such as business communications--relying on our abilities to teach writing for any context, and doing that job fairly well. After all, we are experts on language and linguistics and motivation: we can show people how to use the written word in

CS 209 522

BEST COPY AVAILABLE

order to get something done or to get someone to do something. We should know just how language works and how to use it to delight, to instruct, to move, to express, and to think on paper. As teachers we are experts on how to give students the opportunities to grow as writers--for we have the talent and the training to analyze written discourse in terms of effectiveness and structure. We then share with students our analysis and ask them to imitate those effective structures, and, when necessary, offer an analysis of their written discourse so that they can profit from constructive criticism. It doesn't matter, sometimes, how much exposure we have had in the specific written discourse that we teach or analyze. I can teach business communications even though I know very little about the business world because, and only because, I know something about the principles of communication and language, and because I know writing well enough to translate the needs of my students and the needs of the business environment into a coherent semester-long course.

I could not, of course, pretend to offer a course in computer programming, although I can claim some similarities between effective structures in writing and in computer programming. The knowledge base is just different. Computer languages have their own syntactic and semantic rules, rules that do not seem, at first, like English.

But I can make the computer pretend that it is also an expert on writing, language, and knowledge. It's possible to train the computer to take on the role of teacher to some extent, so that a mythical student can tap into the computer and learn how the expert writer works. I still have to provide the training to the computer, and I still have to be an expert at teaching the topic of writing, but, to make matters complicated, I must now be an expert on myself. How else can I share with a as yet dumb machine the training, education, knowledge, and intuition about language that I have?

Representing Knowledge in a Computer

BEST COPY AVAILABLE

Medical personnel have provided their expertise to computer programs to diagnose diseases and to suggest treatment. Scientists from a number of related and non-related fields--such as research psychology, acoustical engineering, and game developers--have worked on signal interpretation, trying to show a computer how to respond "intelligently" to sounds. Other scientists have offered their expert skills to show how an automatic computer program can design and plan electrical circuitry based on a given need, or even to show how a computer system should be configured for a particular installation. There are expert systems for geology and protein crystallography--so that a computer program can analyze data and provide a picture of inherent structures in earth and living matter. There are also expert systems that supervise the development of other expert systems!

By what magic do these expert systems work? How can MYCIN and INTERNIST correctly diagnose illnesses and propose medical advice? How can DENDRAL determine the structure of an organic molecule from spectroscopic data?

Most expert systems assume that there are a number of ways to interpret a context and a problem. If a medical doctor wants a computer to share knowledge, then the physician can assume that a physical ailment is a problem to be solved, and that one can follow rules for a diagnosis and for a prescription of treatment. The computer programmed that way will follow the heuristic pathways that the expert doctor would test or examine in order to make that diagnosis--checking for particular symptoms, asking systematic questions of the patient, and requesting specific tests of bodily functions. The physician narrows down the possibilities based on how the patient's system matches the ideal, healthy body. That body is something like a car where the parts serve various functions but work to one end--and the mechanic working on the human machine tries to isolate the ignition from the fuel problem, perhaps checking first to make sure the heart is pumping regularly. This method of analysis assumes that knowledge consists of production rules that connect

BEST COPY AVAILABLE

BEST COPY AVAILABLE

individual functions to one governing body or thing. That is, for any one thing to work properly--be it a car or a human--the parts work in harmony; given disharmony, the expert must examine individual functions to see what isn't working properly and why. The expert system based on the notion of production rules would follow the actual steps, assuming that the expert human has trained the system logically and intuitively, that the expert would take.

Another method, though not that dissimilar, is defined by procedural representation of knowledge. Here the heuristic pathways that an expert may take are shown by example as much as by format. While the production rule-based system assumed that there was a problem with some object or human, the procedural-based system may, in fact, indicate the proper functioning of that being or thing. This expert system answers the question "How does it work normally?" Or, "What are the normal procedures operating here?" The expert human and the expert system would still be able to analyze how given data fit the context of the expected, ideal model. Procedural representation of knowledge is also termed "forward chaining," where the expert begins with the known body of facts about function and structure and derives rules that obey the given set of facts. Forward reasoning mechanisms sift through a database of knowledge and find rules that fit these facts--not unlike inductive logic.

Systems based on production rules can be termed "backward chaining" logic machines. Here the expert machine or human starts with a theory or hypothesis about function or structure, looks for a rule already in its set of rules that may answer to a particular problem, and then tries to find a relevant fact that suits the rule--not unlike deductive logic.

A third possible mode of reasoning comes from frame representation theory. Here the assumption is that we can train a computer to "think" associatively. A frame system describes the relationships between objects in terms of goals, potential activities, and locations. Any object in space has a particular set of attributes given it by its relationships to other objects in space. Let me use an example by Earl Hunt. You've been isolated in a cubicle

BEST COPY AVAILABLE

with a table and a few tools, and you face a small window through which objects can be passed. Someone on the other side of that window hands you a number of parts that you must assemble into one object. She gives you what appears to be a wing from an airplane, then the tail section, but in no particular sequence. You probably assemble this object based on your expectations about an airplane's structure and on where the parts should fit. You don't have an instruction manual, nor do you need one, because you can figure out where the pieces belong. You are operating on the principles of frame representation: given clues to a context, you assume that parts belong to an imagined structure according to the expectation that these parts should have definite goals and locations so that the whole object can serve one general purpose. An airplane can only fly with at least two wings, so make sure that you have two wings, a fuselage to attach them to, a tail section, and so forth, before you begin flying your model airplane. (Discussion adapted from Hellwig 77-79 and from Hunt 429-30.)

We use frame representation to react to situations and contexts. In a restaurant we expect to pay for our food and service at the end of our meal; in a fast food place we must pay in advance. A traditional play contains characters, plot, and scenery on a stage--and we participate in that play by purchasing our tickets before we march in at an appointed hour, by sitting in assigned seats (usually), by listening to the play attentively, and by getting up and sitting down at scheduled times. In the restaurant and at the play we obey the expectations of the moment--we don't expect to be served food at the play, nor should we demand a Chekhov play at the restaurant. But both contexts have scripts that we follow--and at a dinner theater, we change the conventions to accommodate a new script.

An expert system using frame representation theory contains packets of information about objects or ideas. Given the classic "most elephants are gray" discussion used by frame theorists, one can see how data is organized. An elephant, as a word in a knowledge base, has attached to it the qualities of

BEST COPY AVAILABLE

grayness, size, intelligence, large floppy ears, trunk, and so forth, qualities that are described as expected values for an elephant. Any time the expert system receives the word "elephant" the computer automatically expects the context to be elephant-oriented; that is, the elephant can't talk, but it can bellow. With this system of representation the data base must be described in terms of qualities and relationships. There are rules, yes, but the emphasis lies in the dormant goals, expectations, and possibilities for objects or things in space.

Expert Systems in Use

DENDRAL, as indicated earlier, determines the structure of organic molecules--the geometry of the molecule, atom bond lengths and angles--using general knowledge, a human expert's heuristics given a similar problem, and any short cuts generally used for interpreting the data from mass spectrometry experiments. DENDRAL is a rule-based expert system, based on backward chaining reasoning and production rules. The inherent problem defined by its use, of course, is that an unknown structure exists, with some data provided by chemical and spectroscopic analysis. Meta-DENDRAL, developed later, evolves rules based on the data received, using inductive logic or forward chaining reasoning mechanisms. Meta-DENDRAL updates DENDRAL's information-handling facilities, though it begins with a similar rule-based system.

The Digital Equipment Corporation designed an in-house expert system that operates as an intelligent instruction manual for engineers who must install the VAX line of computers. Each VAX computer had to be configured or adapted to the location for the computer and to the customer's requirements. Cables and parts had to be installed correctly, in the right order, adjusting to the needs of the installation itself--for example, cable lengths, determined by the computer room and by the customer's placement of terminals, had to be measured in terms of resistance, capicitance, and other electrical values, and

installed to optimize the correct functioning of the computer. These electrical values are significant for mainframe computers; personal computers are less affected by changes in installation equipment values. XCON, once called R1, is again a rule-based expert system, but uses forward chaining reasoning mechanisms as well, deriving new rules for the engineer to follow given a new set of facts concerning the VAX environment and the installation environment (Michaelsen 306-7, McDermott).

Expert systems that use frame representation schemes are fewer in number, principally because they are more complex in design. Steamer, developed jointly by Bolt Beranek and Newman Inc. and the Navy Personnel Research and Development Center, is a tutoring system designed to show the operating procedures used in shipboard steam plants. Procedures and component parts are taught in terms of frames on a screen, showing the student how to operate the shipboard steam plant in terms of the relationships between objects and the functions they play. For example, a computer-generated icon or graphic picture isolates the roles for each part--before you can start the main condensate pump, you need to open the suction valve, start the motor-driven pump and align various valves. The student follows procedures by attempting to run the plant according to general operating principles, but is re-directed by the screen's images to specific steps necessary--and given the inherent connections for each step (Michaelsen 309-10).

Most of these expert systems, briefly mentioned and described here, work on mainframe computers only. Some were developed in the late 1960's and early 1970's--like DENDRAL, MYCIN, INTERNIST, CRYSALIS (used in protein crystallography to determine molecular structures from electron density maps), and a wide variety of scientific expert systems for such fields as geology, ocean surveillance, molecular genetics, space exploration, and symbolic mathematics--to name only a few. The current trend is to develop "shells" that do not contain rules dependent on certain information, that is, what's termed "domain-independent systems." These expert systems allow the user to

construct a knowledge base of one's own and a rule-based set of rules. EMYCIN, for example, attempts to extract the logic used by physicians from MYCIN's structure and to remove the knowledge base (on infectious diseases), in order to show the kinds of reasoning mechanisms used by physicians and in order to construct other expert systems (Lemley 108-112, Michaelsen 303-12).

In fact, given the interest in personal computers and in expert systems, the very current trend is to develop "shells" that can work on personal computers. There are, to be sure, many expert systems on the market today that still concentrate on particular domains, like TAXADVISOR, but a number of developers are trying to produce expert systems that allow the user to construct the rules and the knowledge base. EMYCIN, for example, has now been translated (and reduced in power) by Texas Instruments to a development package called Personal Consultant. Other shells include IBM PC oriented programs such as M.1, Expert-Ease, the Insight Knowledge System, The Intelligent Machine Model, the EXSYS Expert System Development Package, REVEAL, ES/P Advisor, and Lightyear. Most of these shells incorporate text editors, program monitors, reasoning and linguistic mechanisms that interpret data--and, generally, stiff prices (Kinnucan 16-20; see the April 16, 1985 issue of PC Magazine for reviews on Expert Ease, EXSYS, REVEAL, ES/P, TIMM-PC, M.1., and Lightyear; see the April 1985 issue of BYTE on expert systems and a review of Insight).

One system, Knowledge Engineering Environment (KEE), developed in 1983 by IntelliCorp, is probably the best shell on the market today. But it fits only on a LISP machine, a specialized computer that can cost as much as 100,000 dollars. As a package, it uses backward, forward reasoning mechanisms, and a frame-based representation of knowledge. It can be adapted to any field--presently, it has been used by NASA for Space Shuttle flight control, by Lawrence Livermore Laboratory to tune their mass spectrometer, and is being used by Ford Aerospace to develop an expert system to diagnose problems in communications satellites. But it does cost 50,000 dollars in

BEST COPY AVAILABLE

terms of software and initial support. For a description of KEE and its potentials, see Paul Kinnucan's "Software Tools Speed Expert System Development" in the March 1985 issue of High Technology.

The micro-computer packages are far less sophisticated, and far less costly. Insight can be had for an IBM PC for 95 dollars, but lacks the aids one needs for complex rule-based systems, while M.1 costs 10,000 dollars, and provides a serious environment for expert system development. For a review of a number of expert systems, see the April 16, 1985 issue of **PC Magazine.** All of the current crop of expert systems for IBM PC and other microcomputers are somewhat undeveloped--other projects under way will, no doubt, surpass these first expert system shells in terms of price and usefulness.

My Expert System--The Representation of Syntax and Semantics

Briefly described, most computer programs that analyze sentence and word structures are based on procedural representations of knowledge. These systems study natural language (any human language) according to how most syntactic and semantic patterns are used by linguistically competent users of language ("competency" is used as a linguistic term here--shared linguistic traits and abilities). That is, the programmer asks the linguist for rules that govern the language and then encodes these rules into the program. Given a text, the computer program tries to fit these words into assignable categories and structures according to these rules. These forward chaining devices are commonly used to prepare production-based rules for diagnostic programs in linguistic analysis and feedback systems. If it is to be used as a diagnostic tool, the production-based system assumes that all of the text to be read must fit an ideal model of a given utterance, and, if not found, determines possible responses for that text.

In frame-based systems, relationships between objects and the expectations for these relationships, as discussed earlier, are paramount.

A computer program based on frame representation theory is often termed a "semantic grammar" or "case grammar" program. (For the purposes of my own computer program and for simplicity's sake here, I will assume that case grammar-operated programs are the dominant mode of linguistic analysis.) See Avron Barr and Edward A. Feigenbaum's edition of **The Handbook of Artificial Intelligence** Volumes 1-3, Los Altos: William Kaufman, 1982, for a full discussion of Artificial Intelligence research and their Chapter IV, Volume 1, for a good overview of parsing techniques.

I developed a computer program that pretends to act like an expert system in terms of syntactic and semantic patterns in written discourse. I thought at one time that this program was an expert, but I have since decided that the production rules used are not at all transparent nor user-friendly as they should be. I will discuss the ideal features of an expert system in linguistic analysis and interpretation--as well as the ideal features that should be incorporated into the composition classroom--in my final section. Here let us put to the side my own qualms and see what I have done.

First, a bit of theory and background seems in order. I used a semantic differential scale as developed by Charles Osgood and implemented by David R. Heise in order to determine the spatial concepts attached to various English words. I wanted to provide a frame representation for semantics. I also used case grammar arguments as discussed by Charles Fillmore and others in order to determine the spatial relationships in syntactic structures, basically in terms of the narrator's involvement or non-involvement with the created fictional environment. I wanted to assume that I could analyze the logical frames and scripts in syntax that underlie English texts. I used Pascal as my computer language, an IBM 3033 mainframe, the resources available at UCLA, and several years of my life--to work on this program in order to complete my dissertation on Mark Twain.

Second, allow me to explain the semantic differential scale. This scale assumes that words contain a number of spatial concepts, the main ones

BEST COPY AVAILABLE

being evaluation, potency, and activity. Each word contains the same set of connotative values but in different degrees. For example, someone can experience "anxiety," a somewhat negatively charged, and passively accepted word, but if the word used is "terror," then the levels of negativity, power, and activity rise for the reader. These levels were quantified by testing thousands of subjects in terms of their reactions to words, word-pairs, and concepts. After intial testing indicated that evaluation, potency, and activity levels were the main concepts behind each word--how active is "red" compared to "white"? how favorable? how powerful?--David Heise profiled the most common 1,000 English words, excluding function words. The word "agree," for example, has a numeric score of 0.26 for evaluation, -0.90 for activity, and -0.28 for potency. On a scale of -3.00 to 3.00, which models positive scores as "good" or "active" or "tough," the word "agree" seems rather limp in activity and power. The word "happy," in contrast, received an overall rating of 1.64 for evaluation, 1.79 for activity, and -0.29 for potency--a "better" word, yes, and livelier, too, but much less powerful than the word "agree."

My program might just locate a text using these words, with one dominant over the other--this text might be very agreeable on the surface--but decide that the text revealed a rather passive and indifferent attitude, one that might be a bit threatening. The text using "happy" frequently might reveal an active and agreeable attitude, perhaps, but also show an attitude that was a bit taxed and weak. A text using both words in equal frequency would show some kind of balanced view of life, perhaps. It would depend on the combination of words used and how often each appeared.

Next, in terms of syntax, allow me to explain what a case grammar analysis does in general terms. A case grammar study unlayers the logical functions for noun phrases as determined by linguistic convention and how sentence predication modifies meaning. Predication consists of a verb and a number of noun phrases which show notions of Agency, Instrumentality,

Experiencer, Source, Goal, Path, Location, Possession, Receiver, and Force (among others). Role relations determine what function a noun phrase can play in a sentence. Normally, one cannot say meaningfully that "lightning willingly hit the box," because a Force has "no volition," while "I willingly hit the box" can exist as a linguistic utterance because an Agent does have responsibility and volition in actions.

Elizabeth Traugott, as a literary critic using case grammar principles, discusses **Moll Flanders** in terms of the character's inability to take responsibility for her actions. The logical arguments to the narrative voice in that novel indicate that Moll would rather be an Experiencer, Possessor, or Patient than an Agent. The irony in the novel shows the religious hypocrisy Moll exhibits (and which, presumably, Defoe uses to exemplify contemporary mores). Moll's final conversion seems entirely false, because she could not act genuinely, being deprived of the power to act, and being content to remain an Experiencer or Possessor. Traugott uses this analysis to indicate which edition of the text might have been revised by someone other than Defoe, depending on the changes to the later texts which show Moll as an Agent-motivated narrator (211-19).

I was concerned mostly with Experiencer and Agent arguments in a text, in order to decide on Twain's relative trust in his narrator in much the same vein of irony. I also depended heavily on the use of prepositions and verbs as logical determinants in noun phrases. An Experiencer uses few active verbs and many prepositional phrases that indicate Location, as in "I was in London at the time." An Agent might well write "I made my home at London then," showing an active interest in the city as well as indicating London to be a Path along life's road. Prepositional phrases generally represent role relations--"by" can indicate a conveyance, beneficiary, or instrument; "into," a location, a path; and so on. The kind of prepositional phrase could also pinpoint the kind of narrator, given the proper heuristic procedure.

All of this so far was based on frame representations of knowledge.

BEST COPY AVAILABLE

The syntax parser that I developed in order to approach the frame analysis was procedural, dependent on rules derived linguistically, sometimes ad hoc, and sometimes according to a parsing tool termed an augmented transition network system (or ATN). In some loose way, the ATN could be aligned with frame representation theory, but let's stick with the production rules given. Given a central list of words and word roots, the parser looks for words that match the computer dictionary and decodes the text according to what is known. Function words, such as articles, prepositions, and pronouns, and major nouns and verbs are tagged in an initial pass at the sentence. Scores derived from successful matches to the semantic differential scale dictionary are also tabulated automatically at the same time. The parser hunts for a subject of the sentence, defined as a noun phrase with a determiner, a modifier, and a main noun--searching, for instance, for an adjective or a noun after the determiner until it finds a noun or a verb. If successful in that search, then the ATN labels the noun phrase as a potential subject (even if there is no determiner or adjective), and looks for the verb, whether simple, transitive, intransitive, or auxilliary. The ATN parser moves through a number of heuristic pathways, looking for pre-determined syntactic clues, labelling noun phrases in terms of subject and object. The ATN--and I won't describe the process here--can locate complement-embedded sentence patterns and a variety of other phrases.

A similar ATN on a more global scale then re-examines the sentence in terms of case grammar arguments. With the right heuristic in place for noun phrases, pronouns, and kind of verbs being used, the case grammar ATN deciphers the narrator's involvement with the fictional universe. Let's use a very brief example concerning the use of the preposition "under" and verb choice. If "he is under the table," then "he" should be labelled as an Experiencer, because the verb is passive and the prepositional phrase is a Location; if the verb had been active (the subject animate), then the arguments change to Agent/Path. (There are about six hundred and forty

pathways that can be taken, given a subject, verb, object, and prepositional phrase--for each of the forty-three prepositions that I used.) Along the way, of course, various statistical measures are taken so that chi-square analysis, standard deviation scoring, and raw counts can be done later after each sentence analysis (and each "chapter" analysis) of case grammar arguments.

Statistical measures, case grammar "maps," and sentence "maps" (where the subject, object, verb, parts of speech are tagged symbolically) are printed automatically at the end of each sentence analysis. The program also suggests what can be described about the chief narrator's level of faith in what he or she says. (There's a fairly complex algorithm or mathematical model for comparing the average semantic differential scores with the dominant case grammar arguments--one I won't describe here--that a research psychologist has termed "amusing and unprovable and possibly effective.") Here's an example of one sentence analysis:

```
Unlimited power is the ideal thing      JN=AJN
when it is in safe hands.               <S=PJN

Case Grammar Map:  --S,--S,-EXP, LOC,-S
Evaluation Standard Deviation Score: 0.936
Potency Standard Deviation Score: 0.918
Activity Standard Deviation Score: 0.907
Evaluation Score: -0.22
Potency Score: 0.45
Activity Score: 0.24

The Semantic Differential Scale Average suggests
        The Experiencer
     Exhibits--
        Somewhat uncertain
        Somewhat powerful
        Somewhat active
                --Acceptance
The semantic content seems undercut by the case grammar structure.

Dominant arguments for this sentence include:
1 Location(s)      2 Experiencer(s)       3 Source(s)
```

The closing set of analytical comments for each section of text replicates the general format of the sentence by sentence output, but extends the data available: a final sum of the kinds of parts of speech; raw totals for words, sentences, and averages for words per sentence; a final count for words matched against the semantic differential scale dictionary, with an average for

BEST COPY AVAILABLE

words found against total words in the text; a sentence histogram to indicate the relative length of sentences; chi-square scores, standard deviation scores, and overall means for evaluation, potency, and activity levels; a final count of the case grammar arguments; a dignostic comment similar to the sentence-level analysis that suggests how the case grammar argument for the narrative voice relates to the overall average scores of the semantic differential scales; and, finally, a warning to interpret carefully the data produced.

Some brief comments. I have had to simplify drastically the actual workings of the program--I could not detail how the program chops endings off words in order to locate gerunds, adverbs; nor could I show how the program "guesses" at some case grammar arguments. The program is a combination of frame-based, procedural-based, and production rule-based systems, and never seems elegant enough, looking like a tangle of weeds.

The warning to myself as a user should serve all users of any linguistic computer program, because the tendency to accept the analysis with blind faith can be tempting. Although the success rate for correct parses of a text is high--roughly 80%--the notable failures at fathoming some sentences for my program is awkward, at best, to defend. That is why the program generates enormous amounts of paper, so that I do have a sentence map and a separate analysis of each sentence. I needed to analyze by hand the results obtained to make sure they were right. The dissertation was a work in progress, an attempt to begin the development of some kind of automatic parsing tool that could examine the narrator's relative sincerity within a fictive construct. It did work, but it is a rather narrow program in scope. It lacks a context analyzer, so that it could discern differences in discourse genre. It lacks the memory to recall patterns in the text or inherent structures in narrative voice. And it did take a very long time to design and to program.

Potential in the Classroom

BEST COPY AVAILABLE

BEST COPY AVAILABLE

The expert system in the classroom should be flexible and "transparent," so that changes can be added by the instructor in terms of adding new rules or changing the overall control to the system. Expert systems, especially the ones now being developed as shells, often have to be updated and modified to account for new trends and new ideas. In writing pedagogy it seems readily apparent that expert systems must be tailored for different classes and different instructors' styles of teaching.

The expert system in composition should also be similar to the kind of knowledge representation followed by writers. If we write in chunks of information, or if we use nutshelling, then we should teach with an expert system that uses frame representation. If we need to teach sentence combining, or to show the uses of subordinating conjunctions, then an expert system based on rules might be appropriate. If anything, the expert system must be consistent with the kind of knowledge being presented and follow that mode of thinking systematically.

The expert system should also be "user friendly," so that any person using the computer can interact with the expert system on some kind of "personal" basis. Positive feedback, given as immediately as possible, invites the best results for learning. The expert system should also be able to show its own logic for its responses.

Remember, the expert system thus developed should be able to analyze a written discourse, to interact with a student in terms of response and helpful dialogue, and to show the forms of writing that exist for that kind of written discourse. If, for example, this is to be an expert system that teaches the principles of business communication, then I will have to tell the computer how to look at a collection letter--to provide discourse analysis in terms of what the collection letter should do, how it does it, and the format for a collection letter--how to examine the the student's work in terms of the

qualities that a collection letter should have--to give the student meaningful feedback on how well the student performs in terms of this collection letter--and how to instruct the student on the forms of the collection letter--to offer a rationale for each kind of collection letter, whether it be a reminder notice, a discussion letter, or an urgency message. This is a tall order. Do I really have to show the computer how to take into account the psychological advantages of a linked series of motivational letters and messages--moving from gentle reminder to urgent demand--to collect an old debt? Should this expert system be that good on discourse analysis? Can it really respond to concerns of writing when it analyzes a text?

Of course not. The computer, until some mysterious "they" perfect the "silicron," a silicon-based neuron--and it's now technologically feasible--will still react as any digital computer should. It will have two states of existence, a yes and a no. It won't have the intuitive capabilities that we have to judge a piece of writing on the basis of potential rather than on the basis of imitative exactness. The computer would be hard pressed, indeed, to provide an intelligent discourse analysis of a reminder notice that exists as a cartoon, often an effective form of communication for that form of collection letter. It certainly would find it difficult to be an expert on the frustrations of a student who simply hasn't mastered the article "the" yet.

But I am hopeful that for some limited contexts the expert computer system can be a surrogate learner and teacher. It all depends on how well the computer has been trained to handle the ways knowledge can be represented. If we figure out how knowledge can be used by an expert in ways to solve a problem, then we can show the machine how to address a problem in writing or in knowledge--and how to show a student the ways to write around or for a specific context. It's been done for other fields--why not for composition?

BEST COPY AVAILABLE

Works Cited

Barr, Avron, and Edward A. Feigenbaum, eds. The Handbook of Artificial Intelligence. 3 vols. Los Altos: William Kaufman, 1982. Vol. 1.

Derfler, Frank J. "Expert-Ease Makes Its Own Rules." PC Magazine 16 April 1985: 119-25.

---. "An Affordable Adviser." PC Magazine 16 April 1985: 113-19.

Heise, David. "Semantic Differential Profiles for 1,000 Most Frequent English Words." Psychological Monographs 79 (1965): 1-31.

Hellwig, Harold. "Mark Twain's Response to Democracy and Technology: A Computer-Aided Psycholinguistic Analysis of A Connecticut Yankee in King Arthur's Court." Diss. University of California, Los Angeles, 1985.

Hunt, Earl B. Artificial Intelligence. New York: Academic Press, 1975.

Kinnucan, Paul. "Software Tools Speed Expert System Development." High Technology March 1985: 16-20.

Lemley, Brad. "Artificial Expertise: Intelligent Software for Problem Solving." PC Magazine 16 April 1985: 108-13.

McDermott, J. "R1: A Rule-Based Configuror of Computer Systems." Computer Science Department. Carnegie-Mellon University, 1980.

Michaelsen, Robert H., Donald Michie, and Albert Boulanger. "The Technology of Expert Systems." Byte April 1985: 303-15.

Taylor, Jared. "Lightyear's Ahead of Paper." PC Magazine 16 April 1985: 159-73.

Tello, Ernie. "Marching to a Different Drummer." PC Magazine 16 April 1985: 139-45.

---. "Raw Power for Problem Solving." PC Magazine 16 April 1985: 131-9.

Thompson, Beverly A., and William A. Thompson. "Inside an Expert System." Byte April 1985: 315-30.

BEST COPY AVAILABLE

Traugott, Elizabeth Closs, and Mary Louise Pratt. Linguistics for Students of Literature. New York: Harcourt Brace Jovanovich, 1980.

Webster, Robin. "M.I. Makes a Direct Hit." PC Magazine 16 April 1985: 145-59.

---. "Revealing Business Solutions." PC Magazine 16 April 1985: 125-31.

BEST COPY AVAILABLE